Sequence Comparison

T: +27 51 401 9111 | E: info@ufs.ac.za | **www.ufs.ac.za**

UFSUV | UFSweb | UFSweb | ufsuv

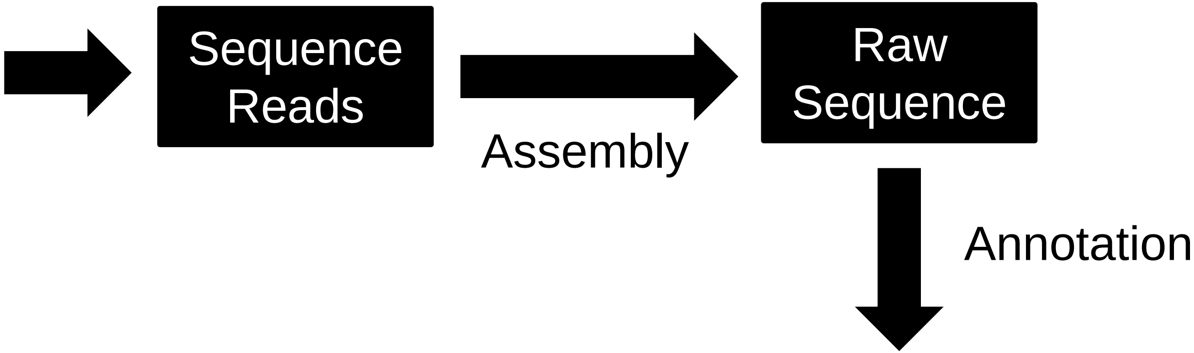
Inspiring excellence.
Transforming lives.

UNIVERSITY OF THE
FREE STATE
UNIVERSITEIT VAN DIE
VRYSTAAT
YUNIVESITHI YA
FREISTATA

UFS
INFORMATION AND
COMMUNICATION
TECHNOLOGY SERVICES
(ICT SERVICES)

Introduction

Sequencing Experiment

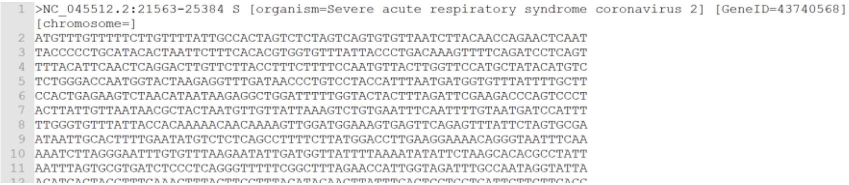


Sequence Reads

Assembly

Raw Sequence

Annotation

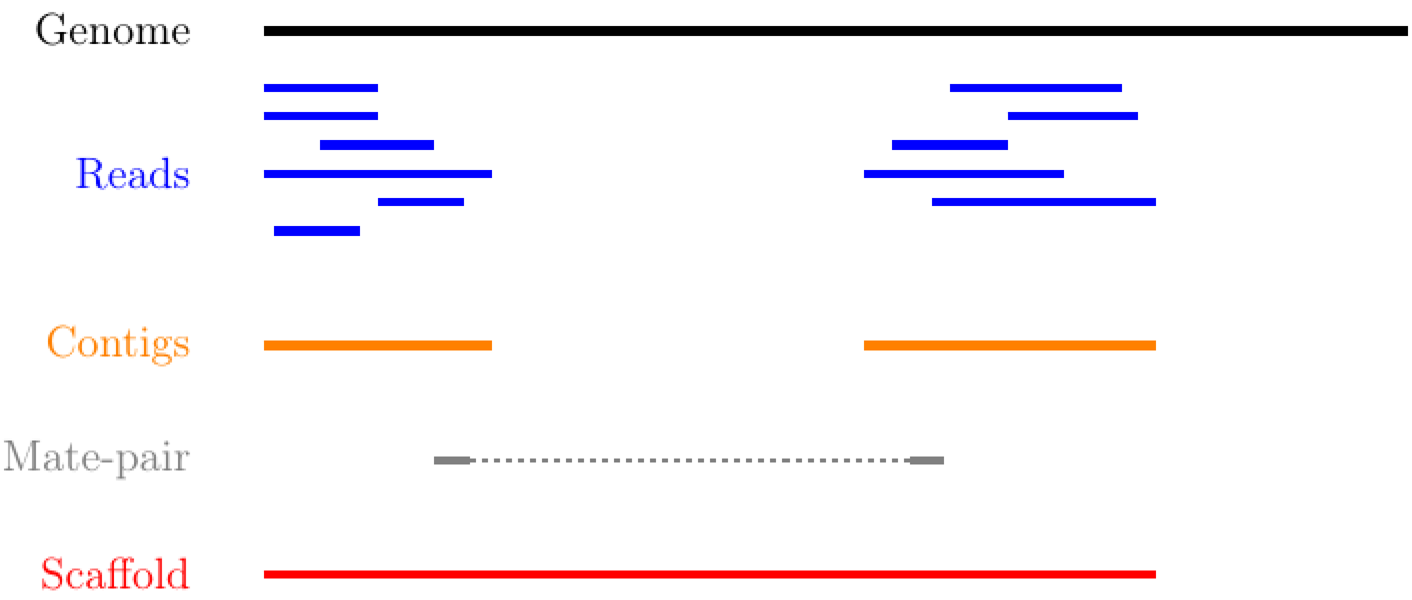


```
>NC_045512.2:21563-25384 S [organism=Severe acute respiratory syndrome coronavirus 2] [GeneID=43740568]
[chromosome=]
ATGTTTGTTTTTCTTGTTTTATTGCCACTAGTCTCTAGTCAGTGTGTTAATCTTACAACCAGAACTCAAT
TACCCCCTGCATACACTAATTCTTTCACACGTGGTGTTTATTACCCTGACAAAGTTTTCAGATCCTCAGT
TTTACATTCAACTCAGGACTTGTTCTTACCTTTCTTTTCCAATGTTACTTGGTTCCATGCTATACATGTC
TCTGGGACCAATGGTACTAAGAGGTTTGATAACCCTGTCCTACCATTTAATGATGGTGTTTATTTTGCTT
CCACTGAGAAGTCTAACATAATAAGAGGCTGGATTTTTGGTACTACTTTAGATTCGAAGACCCAGTCCCT
ACTTATTGTTAATAACGCTACTAATGTTGTTATTAAAGTCTGTGAATTTCAATTTTGTAATGATCCATTT
TTGGGTGTTTATTACCACAAAAACAACAAAAGTTGGATGGAAAGTGAGTTCAGAGTTTATTCTAGTGCGA
ATAATTGCACTTTTGAATATGTCTCTCAGCCTTTTCTTATGGACCTTGAAGGAAAACAGGGTAATTTCAA
AAATCTTAGGGAATTTGTGTTTAAGAATATTGATGGTTATTTTAAAATATATTCTAAGCACACGCCTATT
AATTTAGTGCGTGATCTCCCTCAGGGTTTTTCGGCTTTAGAACCATTGGTAGATTTGCCAATAGGTATTA
```

Sequencing Experiment

- Sequencer / Sequencing technology produces sequences
  - Sanger / Illumina / Roche 454 / PacBio / Oxford Nanopore / etc
- NGS → Many short reads (sequences)
  - Ex. Illumina → 150 bp per read

Sequence Assembly



Sequence Annotation

- Sequences unknown
- Annotation involves:
  - Finding Genes
  - Finding elements. Ex. CPG Islands, Transcription Factors, etc.
- Determine sequence identity
  - **Infer identity by comparison with a known sequence reference**

Sequence Comparison

- Essential step in structure / function analysis
- Lies at the core of bioinformatics analysis!
- How do we compare sequences?

Pairwise Sequence Alignment

- Process of comparing two sequences to each other
  - Search for common patterns
  - Search for per residue correspondence
- Forms the basis of:
  - Database Similarity Searching
  - Multiple Sequence Alignment
    - Homology Modelling
    - Phylogenetic Analysis

Pairwise Sequence Alignment

ATGGGAACCTCCG

AACCTCCGTAAAA

Pairwise Sequence Alignment

```
ATGGGAACCTCCG
     AACCTCCGTAAAA
```

Evolutionary basis for sequence similarity

- Protein and DNA sequences are products of evolution
- Sequences will change over time
  - Random mutations / insertions / deletions
- Some sequences will be preserved by natural selection
  - Particularly sequences crucial to structure and function
  - We can use these “traces” to identify common ancestors
- Degrees of sequence conservation reveals evolutionary relatedness
- Degrees of variation reveals evolutionary divergence

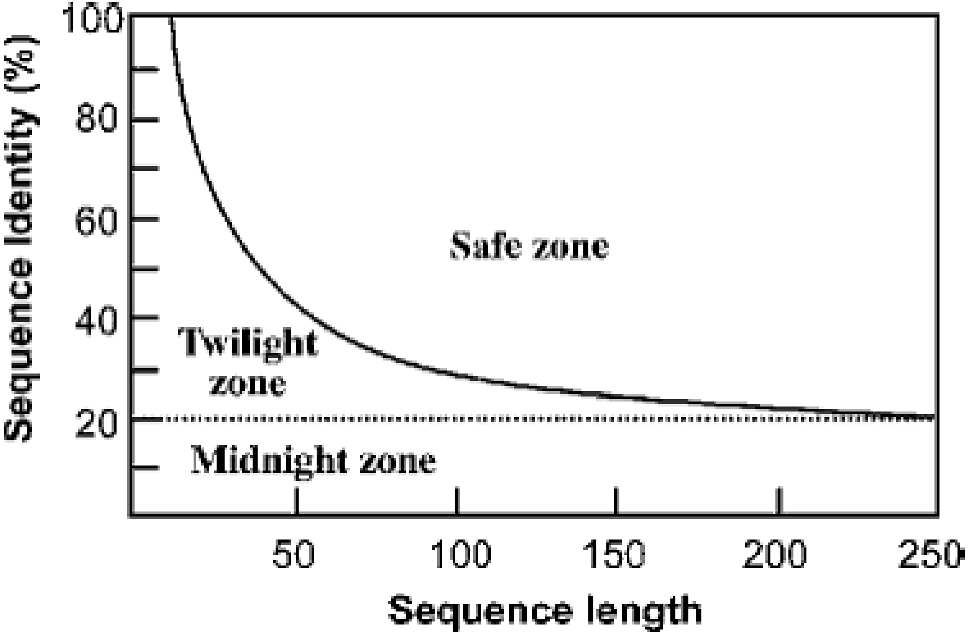
Sequence similarity vs Sequence homology

- Sequence A is homologous to Sequence B
  - A and B share a common ancestor
  - Binary classifier : Homologous or nonhomologous
  - I.e. No such thing as 40% homologous sequences
- Sequence similarity
  - Literally how similar A is to B
  - Example: A = DAG and B = DPG
  - Sequence similarity = ~66%

Sequence similarity – Random Matching

- Sequence matches can be random
  - Nucleic Acids : 25% chance of a random match (1/4)
  - Amino Acids : 5% chance of a random match (1/20)
  - Introduction of gaps → Rises chance of random matching by 10 – 20%
- Sequence length is important
  - Short sequence matches → higher probability of random matching

Sequence similarity – Random Matching



Sequence similarity vs Sequence Identity

- Synonymous for nucleotide sequences
- Amino Acid sequences
  - Identity = Exact amino acid residue matches (A → A)
  - Similarity = Physiochemical matches (K → R)
- Caveat with physiochemical matches
  - Handle with care – the mismatch may have structural meaning
  - Example: Histone Acetyl Transferase (HAT) – modifies a K but cannot modify a R
- Two methods to calculate sequence similarity / identity

Method 1

$$S = \left[\frac{(L_S \times 2)}{(L_a + L_b)}\right] \times 100$$

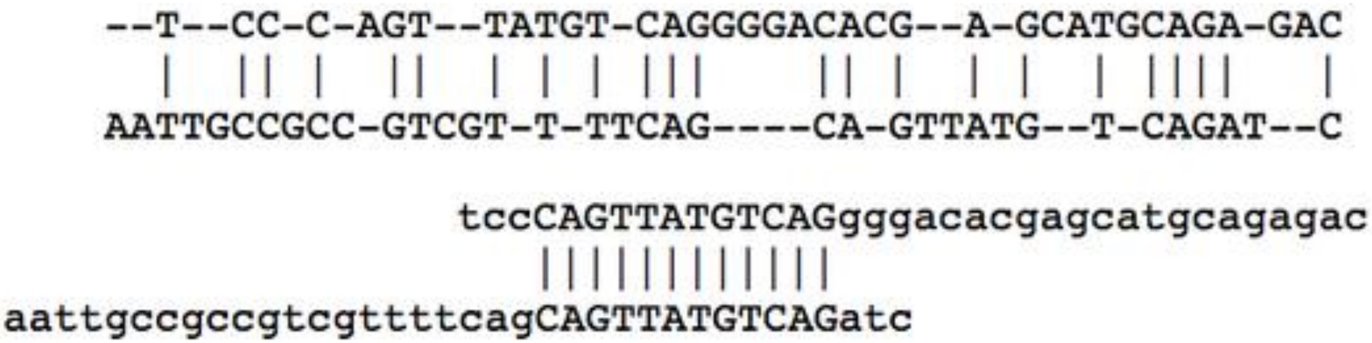
- S = % sequence similarity
- L_S = number of aligned residues with similar characteristics
- L_a, L_b = Lengths of each individual sequences A and B

Method 2 – Normalizing for short sequences

$$S\% = \frac{L_S}{L_a}\%$$

- S = % sequence similarity
- L_S = number of aligned residues with similar characteristics
- L_a = Length of the shortest sequence

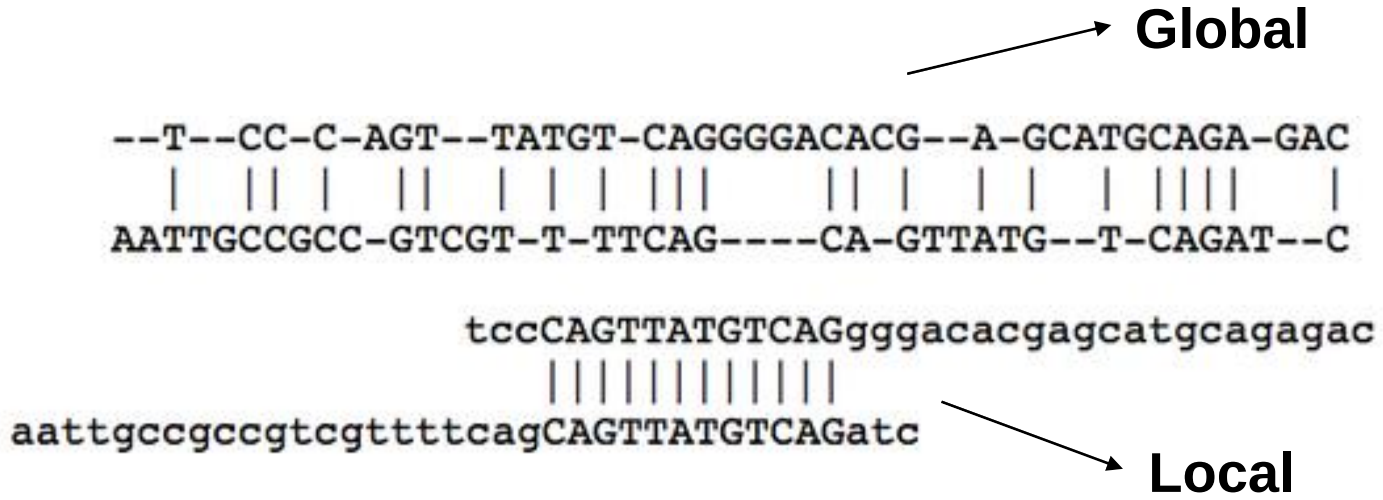
Sequence alignment : Global vs Local



```
    --T--CC-C-AGT--TATGT-CAGGGGACACG--A-GCATGCAGA-GAC
      |  || |  ||  | | | |||    || |  | |  | ||||   |
    AATTGCCGCC-GTCGT-T-TTCAG----CA-GTTATG--T-CAGAT--C

                  tccCAGTTATGTCAGgggacacgagcatgcagagac
                     ||||||||||||
aattgccgccgtcgttttcagCAGTTATGTCAGatc
```

Sequence alignment : Global vs Local



Global

```
--T--CC-C-AGT--TATGT-CAGGGGACACG--A-GCATGCAGA-GAC
  |  || |  ||  | | | |||    || |  | |  | ||||   |
AATTGCCGCC-GTCGT-T-TTCAG----CA-GTTATG--T-CAGAT--C
```

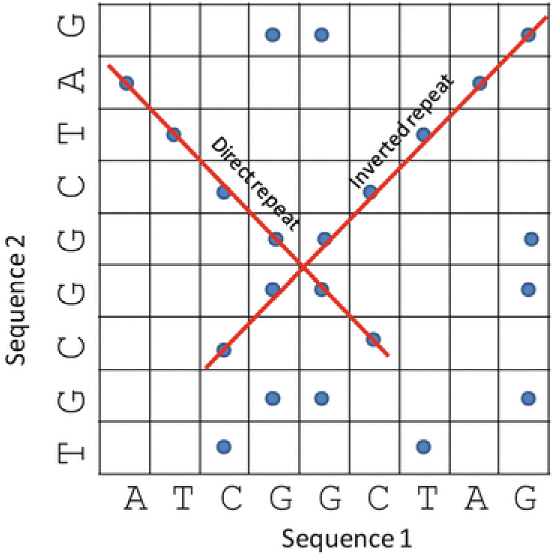
```
                  tccCAGTTATGTCAGgggacacgagcatgcagagac
                     ||||||||||||
aattgccgccgtcgttttcagCAGTTATGTCAGatc
```

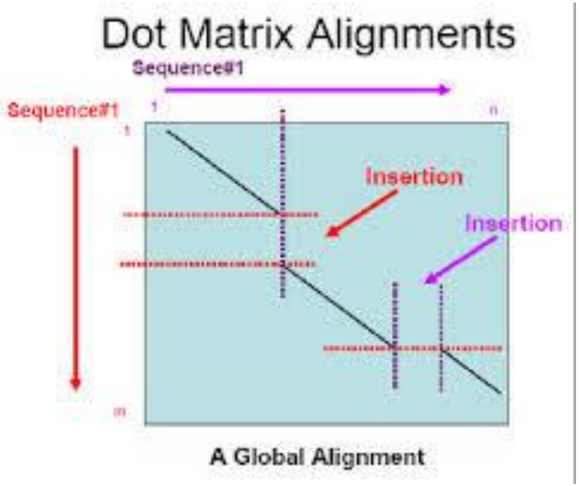
Local

Sequence Alignment Algorithms

- Systematic computer “protocol” to align sequences
- Two main types:
  - Dot Matrix Method
  - Dynamic Programming

Dot Matrix Algorithm





Advantages

- Graphical representation of alignments
- Can easily identify regions of sequence similarity
- Particularly useful for identifying repeat sequences – parallel diagonals of same size (see previous image)
- For nucleic acids – can aid in identification of secondary structures via detecting self-complimentary sequences (Align a sequence with itself)

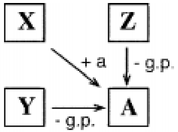
Disadvantages

- High noise level for long sequences
- It displays all matches – user has to assemble the full alignment in the case of insertions and deletions
- Lacks statistical rigor for assessment of the quality of the alignment
- Difficult to scale up to multiple sequence alignment – thus it is primarily used for pair-wise sequence alignment

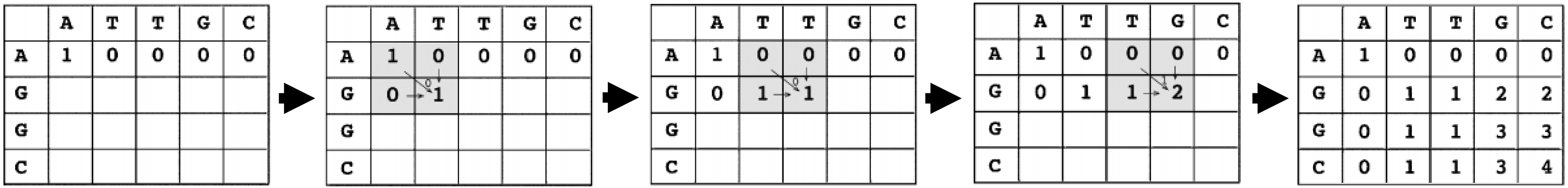
Dot Matrix Sequence Alignment Software

- https://www.expasy.org/genomics/sequence_alignment

Dynamic Programming



A is the maximum score from one of the three directions plus matching score at the current position



	A	T	T	G	C
A	1	0	0	0	0
G					
G					
C					

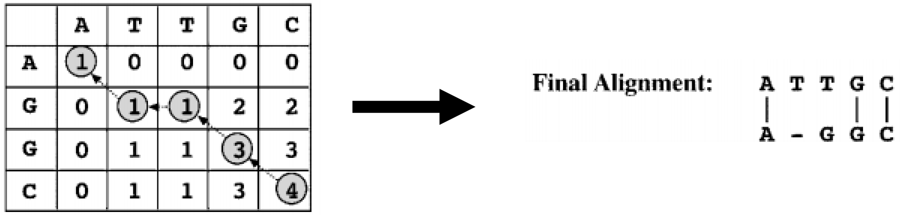
	A	T	T	G	C
A	1	0	0	0	0
G	0	1			
G					
C					

	A	T	T	G	C
A	1	0	0	0	0
G	0	1	1		
G					
C					

	A	T	T	G	C
A	1	0	0	0	0
G	0	1	1	2	
G					
C					

	A	T	T	G	C
A	1	0	0	0	0
G	0	1	1	2	2
G	0	1	1	3	3
C	0	1	1	3	4

Dynamic Programming



	A	T	T	G	C
A	1	0	0	0	0
G	0	1	1	2	2
G	0	1	1	3	3
C	0	1	1	3	4

Final Alignment:

```
A T T G C
|     | |
A - G G C
```

Dynamic Programming

- Brute force method – needs lots of computational resources
- Global alignment – Needleman-Wunsch algorithm
  - Extends from beginning of sequence until the end of the sequence
  - Focusses on best global score – so may miss best local alignments
- Local alignment – Smith-Waterman algorithm
  - Can extend from anywhere in the matrix
  - Focusses on the best regional scores – thus may miss best global alignment

Scoring Matrices

- Dynamic Programming uses a scoring system
  - Set of values for quantifying the likelihood of one residue being a substituted by another in an alignment
- Scoring System is called a substitution matrix
  - Derived from statistical analysis of residue substitution data from sets of reliable alignments of highly related sequences

Scoring Matrices – Nucleic Acids

- Relatively simple
  - Positive value or high score for a match
  - Negative or low score for a mismatch
  - It is assumed frequency of mutation between all bases are equal
- Sources of inaccuracy
  - Transitions (substitutions between purine and purine, and pyrimidine and pyrimidine) occur more frequently than transversions (purine to pyrimidine)
  - Solution : Use more sophisticated Stats models

Scoring Matrices – Amino Acids

- More complex – more amino acid residues than nucleic acid residues
- Two common matrices
  - PAM (Point Accepted Mutation)
    - Margret Dayhoff compiled alignments of 72 groups of very closely related proteins
  - BLOSUM
    - Series of blocks amino acid substitution matrices – Direct observation in multiple sequence alignments

Amino Acid Scoring Matrices – PAM

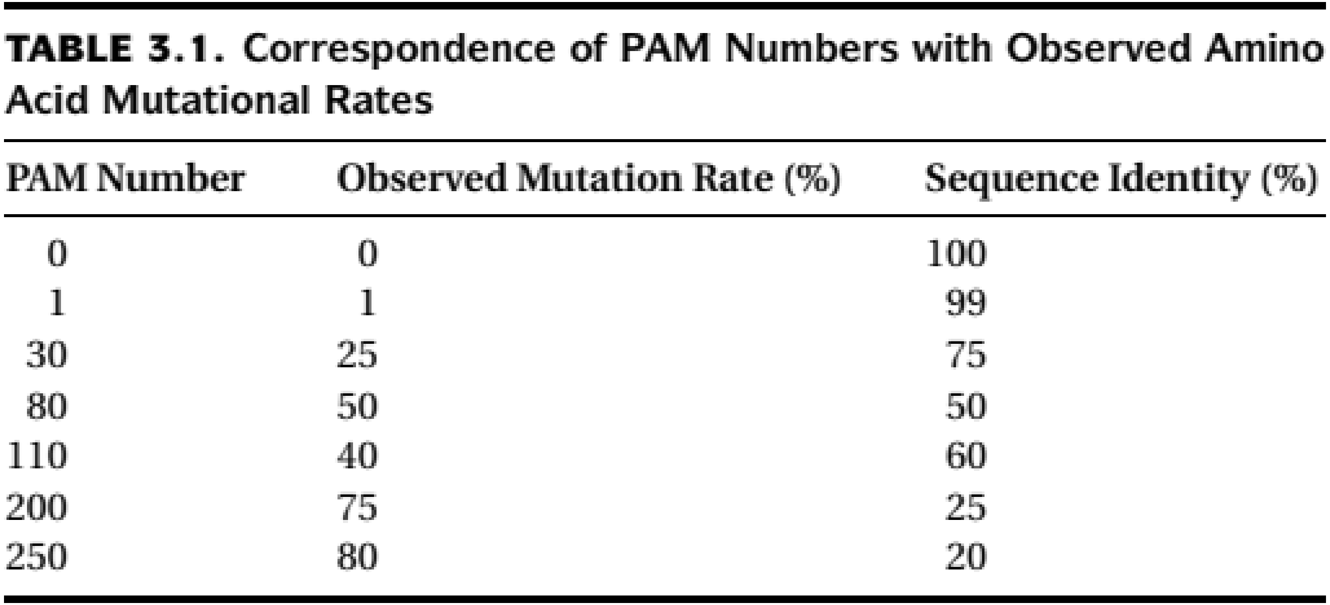
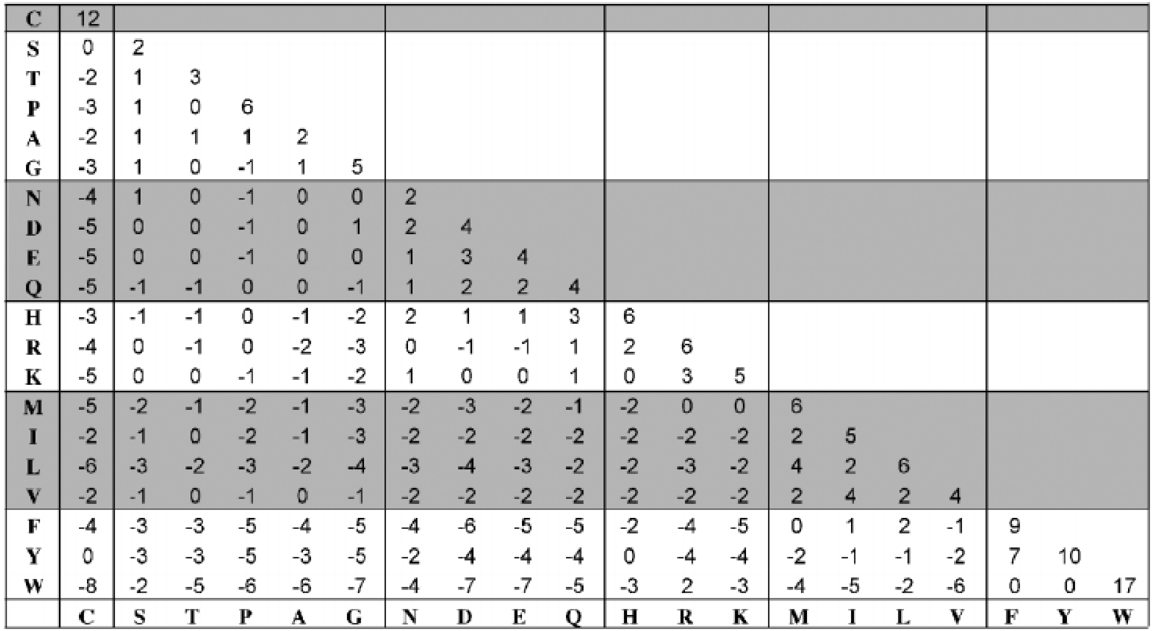


TABLE 3.1. Correspondence of PAM Numbers with Observed Amino Acid Mutational Rates

PAM Number	Observed Mutation Rate (%)	Sequence Identity (%)
0	0	100
1	1	99
30	25	75
80	50	50
110	40	60
200	75	25
250	80	20

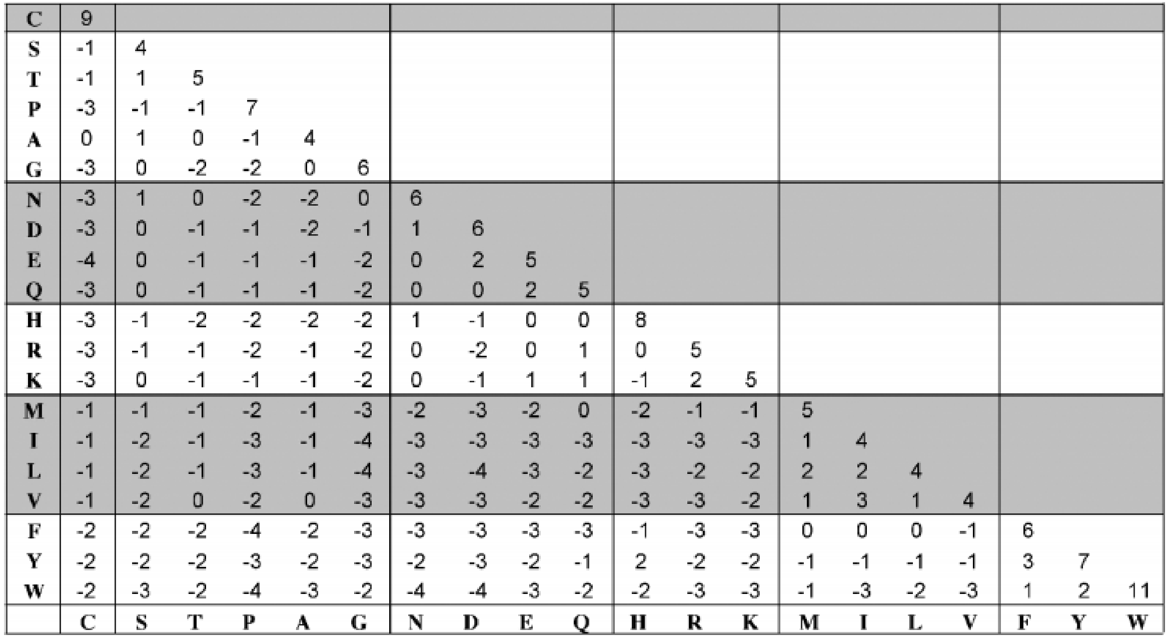
Amino Acid Scoring Matrices – PAM250



C	12																			
S	0	2																		
T	-2	1	3																	
P	-3	1	0	6																
A	-2	1	1	1	2															
G	-3	1	0	-1	1	5														
N	-4	1	0	-1	0	0	2													
D	-5	0	0	-1	0	1	2	4												
E	-5	0	0	-1	0	0	1	3	4											
Q	-5	-1	-1	0	0	-1	1	2	2	4										
H	-3	-1	-1	0	-1	-2	2	1	1	3	6									
R	-4	0	-1	0	-2	-3	0	-1	-1	1	2	6								
K	-5	0	0	-1	-1	-2	1	0	0	1	0	3	5							
M	-5	-2	-1	-2	-1	-3	-2	-3	-2	-1	-2	0	0	6						
I	-2	-1	0	-2	-1	-3	-2	-2	-2	-2	-2	-2	-2	2	5					
L	-6	-3	-2	-3	-2	-4	-3	-4	-3	-2	-2	-3	-2	4	2	6				
V	-2	-1	0	-1	0	-1	-2	-2	-2	-2	-2	-2	-2	2	4	2	4			
F	-4	-3	-3	-5	-4	-5	-4	-6	-5	-5	-2	-4	-5	0	1	2	-1	9		
Y	0	-3	-3	-5	-3	-5	-2	-4	-4	-4	0	-4	-4	-2	-1	-1	-2	7	10	
W	-8	-2	-5	-6	-6	-7	-4	-7	-7	-5	-3	2	-3	-4	-5	-2	-6	0	0	17
	C	**S**	**T**	**P**	**A**	**G**	**N**	**D**	**E**	**Q**	**H**	**R**	**K**	**M**	**I**	**L**	**V**	**F**	**Y**	**W**

Figure 3.5: PAM250 amino acid substitution matrix. Residues are grouped according to physicochemical similarities.

Amino Acid Scoring Matrices – BLOSUM62



C	9																			
S	-1	4																		
T	-1	1	5																	
P	-3	-1	-1	7																
A	0	1	0	-1	4															
G	-3	0	-2	-2	0	6														
N	-3	1	0	-2	-2	0	6													
D	-3	0	-1	-1	-2	-1	1	6												
E	-4	0	-1	-1	-1	-2	0	2	5											
Q	-3	0	-1	-1	-1	-2	0	0	2	5										
H	-3	-1	-2	-2	-2	-2	1	-1	0	0	8									
R	-3	-1	-1	-2	-1	-2	0	-2	0	1	0	5								
K	-3	0	-1	-1	-1	-2	0	-1	1	1	-1	2	5							
M	-1	-1	-1	-2	-1	-3	-2	-3	-2	0	-2	-1	-1	5						
I	-1	-2	-1	-3	-1	-4	-3	-3	-3	-3	-3	-3	-3	1	4					
L	-1	-2	-1	-3	-1	-4	-3	-4	-3	-2	-3	-2	-2	2	2	4				
V	-1	-2	0	-2	0	-3	-3	-3	-2	-2	-3	-3	-2	1	3	1	4			
F	-2	-2	-2	-4	-2	-3	-3	-3	-3	-3	-1	-3	-3	0	0	0	-1	6		
Y	-2	-2	-2	-3	-2	-3	-2	-3	-2	-1	2	-2	-2	-1	-1	-1	-1	3	7	
W	-2	-3	-2	-4	-3	-2	-4	-4	-3	-2	-2	-3	-3	-1	-3	-2	-3	1	2	11
	C	S	T	P	A	G	N	D	E	Q	H	R	K	M	I	L	V	F	Y	W

Figure 3.6: BLOSUM62 amino acid substitution matrix.

Differences between PAM and BLOSUM

- All PAM matrices except PAM1 derived from evolutionary model
- BLOSUM values are exclusively direct observation – may have less evolutionary meaning
- PAM is better for long – closely related sequences
- BLOSUM outperform PAM in local alignments
    - Based on a much larger dataset
- Other matrices
    - Gonnet and Jones – Taylor – Thorton
    - Same performance as BLOSUM but more robust for constructing phylogenetic trees

Which one should you use?

- No clear winner
  - BLOSUM recommended for general use
  - PAM recommended for closely related relatives
- Best way is to try all and compare the alignments
- Also try to pick a matrix derived from sources which closely resembles your subject of study

Database similarity searching

- Main application of pairwise sequence alignment → retrieving matching biological sequences in databases
- What happens when you submit a query sequence for search against a DB?:
  - Pairwise alignment with all sequences in the DB
  - Dynamic programming nor dot matrix alignment algorithms are suited for this!
  - We need a better algorithm

DB similarity searching algorithm requirements

- Sensitivity
  - Ability to find as many correct hits as possible (True positives)
- Selectivity
  - Ability to exclude incorrect hits (False positives)
- Speed
  - Time it takes to search and return results

Searching Requirements – Reality check

- Like the old saying : “Between health / wealth / happiness you can’t realistically have all three”
- Increase in sensitivity → searches too inclusive (greedy) → many false positives
- Increase in speed at cost of sensitivity and selectivity

Algorithm Types

- Exhaustive vs Heuristic

Exhaustive Algorithms

- Rigorous → find exact solution to the problem by examining all possible mathematical solutions
- Dynamic Programming
- Computationally expensive and slow

Heuristic Algorithms

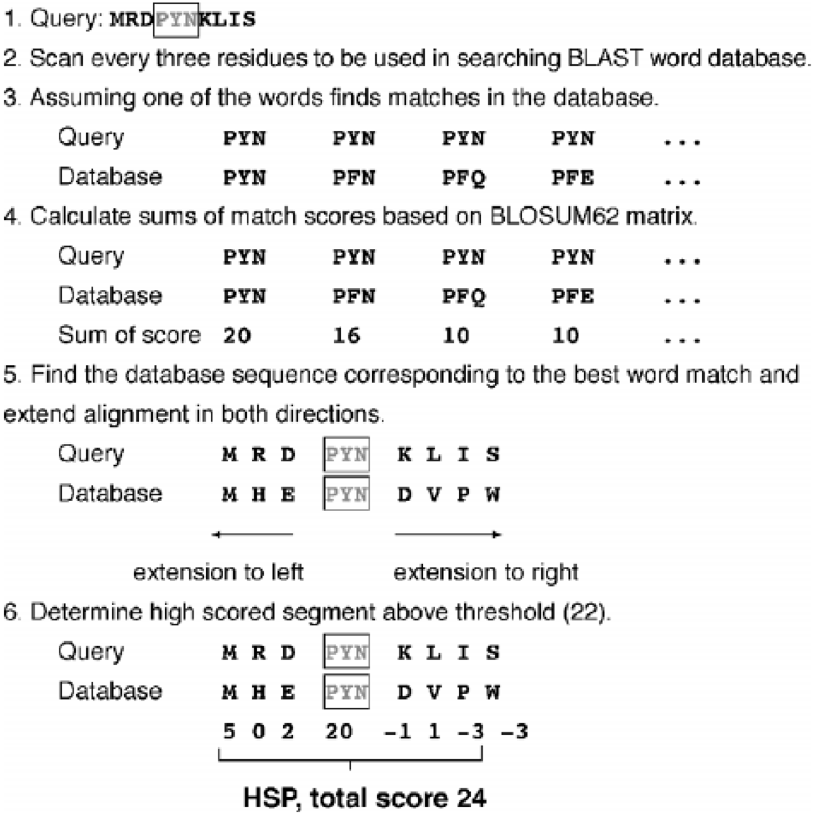
- Computational strategy to find the closest solution
- Generally make assumptions (i.e. take shortcuts) to reduce the search space
- Occam's razor – Simpler solutions are more likely to be correct than complex solutions
- Key advantage - **SPEED**

Basic Local Alignment Search Tool (BLAST)

- Developed by Stephen Altschul @ NCBI in 1990
- Heuristic Word Method to align query sequence to all sequences in a database
- Versus Dynamic Programming Algorithm:
  - 50 – 100 times faster
  - Moderate knock to similarity and specificity

Basic Local Alignment Search Tool (BLAST)

- Objective: Find high-scoring ungapped segments among related sequences
- Existence of these segments above a defined threshold indicates pairwise similarity beyond random chance
- Thus, BLAST discriminates between unrelated sequences in the database



1. Query: MRDPYNKLIS

2. Scan every three residues to be used in searching BLAST word database.

3. Assuming one of the words finds matches in the database.

```
Query          PYN    PYN    PYN    PYN    ...
Database       PYN    PFN    PFQ    PFE    ...
```

4. Calculate sums of match scores based on BLOSUM62 matrix.

```
Query          PYN    PYN    PYN    PYN    ...
Database       PYN    PFN    PFQ    PFE    ...
Sum of score   20     16     10     10     ...
```

5. Find the database sequence corresponding to the best word match and extend alignment in both directions.

```
Query          M R D  PYN  K L I S
Database       M H E  PYN  D V P W
```

6. Determine high scored segment above threshold (22).

```
Query          M R D  PYN  K L I S
Database       M H E  PYN  D V P W
               5 0 2  20  -1 1 -3 -3
```

HSP, total score 24

Figure 4.1: Illustration of the BLAST procedure using a hypothetical query sequence matching with a hypothetical database sequence. The alignment scoring is based on the BLOSUM62 matrix (see Chapter 3). The example of the word match is highlighted in the box.

BLAST Scoring – E-value

- Outputs list of pairwise sequence matches ranked by statistical significance (E-value)

$$\boldsymbol{E = m \times n \times P}$$

- Where:
  - m = Total number of residues in a database
  - n = Number of residues in the query sequence
  - p = Probability that an HSP alignment is the result of random chance

BLAST Scoring – E-value

- Likelihood that a hit is purely by chance
- Thus, the lower the value, the higher the probability that the hit is a true positive
- Empirical implementation:
  - $E < 1 \times 10^{-50}$: Extremely high confidence that the match is result of homologous relationships
  - $1 \times 10^{-50} < E < 0.01$: Considered a result of homology
  - $0.01 < E < 10$: Considered not significant (Additional evidence required if this not the case)
  - $E > 10$: Unrelated sequence

BLAST Scoring – E-value

$$\boldsymbol{E = m \times n \times P}$$

- Proportional to DB size
  - E-value of match will grow as DB grows
  - Genuine matches likely unaffected – but you will “lose” matches
- Alternative : Use the bit-score

BLAST Scoring – Bit Score

- Sequence similarity independent of sequence length and database size
- Normalized on the precise raw alignment score

$$S' = \frac{(\lambda \times S - \ln K)}{\ln 2}$$

- Where:
  - λ = Gumble distribution constant
  - S = Raw alignment score
  - K = Constant associated with scoring matrix
- The higher the bit score – the higher the significance of the match

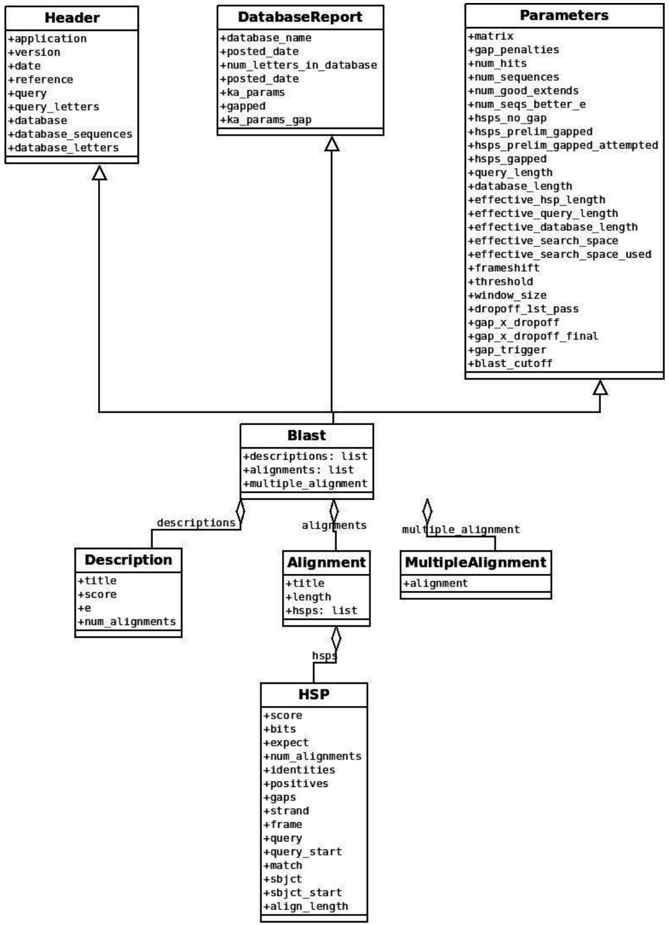
BLAST Variants

Program	Database type	Query
blastn	nucleotide	nucleotide
blastp	protein	protein
blastx	protein	nucleotide translated to protein
tblastn	nucleotide translated to protein	protein
tblastx	nucleotide translated into protein	nucleotide translated into protein

BLAST Output File

- Many output options available
- For portability → Use XML output (Option 5)
  - Most stable file format → Text output formats have a tendency to change
  - Format of choice for downstream parsers

BLAST XML file format



Header

- +application
- +version
- +date
- +reference
- +query
- +query_letters
- +database
- +database_sequences
- +database_letters

DatabaseReport

- +database_name
- +posted_date
- +num_letters_in_database
- +posted_date
- +ka_params
- +gapped
- +ka_params_gap

Parameters

- +matrix
- +gap_penalties
- +num_hits
- +num_sequences
- +num_good_extends
- +num_seqs_better_e
- +hsps_no_gap
- +hsps_prelim_gapped
- +hsps_prelim_gapped_attempted
- +hsps_gapped
- +query_length
- +database_length
- +effective_hsp_length
- +effective_query_length
- +effective_database_length
- +effective_search_space
- +effective_search_space_used
- +frameshift
- +threshold
- +window_size
- +dropoff_1st_pass
- +gap_x_dropoff
- +gap_x_dropoff_final
- +gap_trigger
- +blast_cutoff

Blast

- +descriptions: list
- +alignments: list
- +multiple_alignment

Description

- +title
- +score
- +e
- +num_alignments

Alignment

- +title
- +length
- +hsps: list

MultipleAlignment

- +alignment

HSP

- +score
- +bits
- +expect
- +num_alignments
- +identities
- +positives
- +gaps
- +strand
- +frame
- +query
- +query_start
- +match
- +sbjct
- +sbjct_start
- +align_length

BLAST Availability

- Most common interface:
  https://blast.ncbi.nlm.nih.gov/Blast.cgi
- BLAST+ is the command-line executables distributed via FTP
- Adapted to be accelerated on many hardware platforms
  - GPU
  - HPC's (Across many CPUs and Nodes)
- Part of almost any bioinformatics pipeline